Installation manual for Python & Co.

Release 0.1

Hans-Martin von Gaudecker

February 15, 2011

Contents

1 Preface

This is a little document for preparing you to install all programs that are necessary for the entire course on your machine. It will gradually evolve with your feedback. **Please 'drop me a line <mailto:hmgaudecker@gmail.com>'_ if any links are outdated, descriptions are unclear, or you encounter problems.**

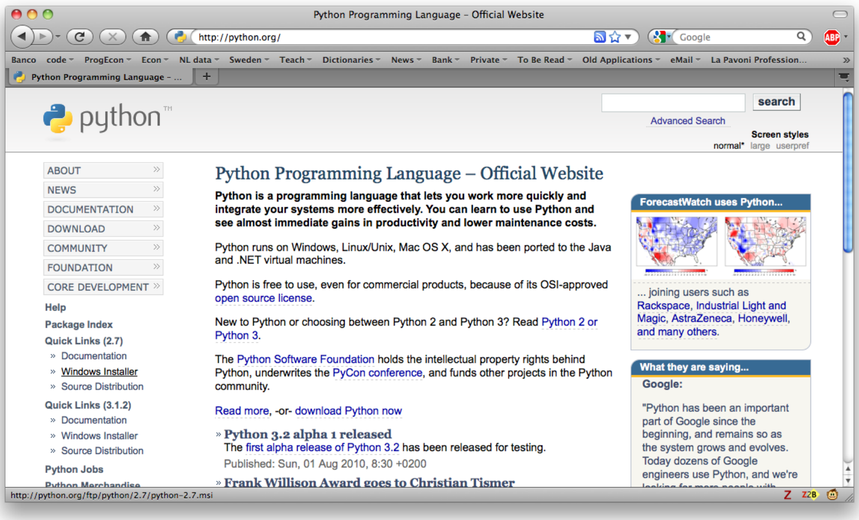
If everything goes smoothly and you have a fast machine and a fast Internet connection, you should be done within half an hour. If you run into any problems or need considerably more time than this, don't hesitate to ask me or one of the IT administrators to help you out. I expect this to be a hurdle for some of you at this point, son don't be shy to ask for help.

Please follow the steps one-by-one in and do not deviate AT ALL from them, unless you really know what you are doing. Getting off the beaten path has proven to be the #1 source of errors in previous rounds.

2 Python and Python libraries

2.1 Python 2.7.1

You first need to download Python, version 2.7.1 This is the main current version, even though the 3.x line is already being developed. However, the latter is not backwards compatible to the 2.x series and many of the very complex scientific libraries are not supported under 3.x yet. 2.7 will be supported for years to come (and already incorporates many of the new features of 3.x), so you will not learn something that is outdated. Get the Windows Installer. If the link doesn't work or you have another operating system, direct your browser to http://python.org/ and click on the left side of the website on *Quick Links (2.7)* >> *Windows Installer* (or whatever is appropriate). If you are on Windows, do **NOT** install the 64-bit versions as the scientific libraries are not available in 64-bit. Just use the above link if in doubt.



Python Programming Language – Official Website

Python is a programming language that lets you work more quickly and integrate your systems more effectively. You can learn to use Python and see almost immediate gains in productivity and lower maintenance costs.

Python runs on Windows, Linux/Unix, Mac OS X, and has been ported to the Java and .NET virtual machines.

Python is free to use, even for commercial products, because of its OSI-approved open source license.

New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3.

The Python Software Foundation holds the intellectual property rights behind Python, underwrites the PyCon conference, and funds other projects in the Python community.

Read more, -or- download Python now

» Python 3.2 alpha 1 released
The first alpha release of Python 3.2 has been released for testing.
Published: Sun, 01 Aug 2010, 8:30 +0200

» Frank Willison Award goes to Christian Tismer

... joining users such as Rackspace, Industrial Light and Magic, AstraZeneca, Honeywell, and many others.

Google:

"Python has been an important part of Google since the beginning, and remains so as the system grows and evolves. Today dozens of Google engineers use Python, and we're

Run the installer, accepting all default options.

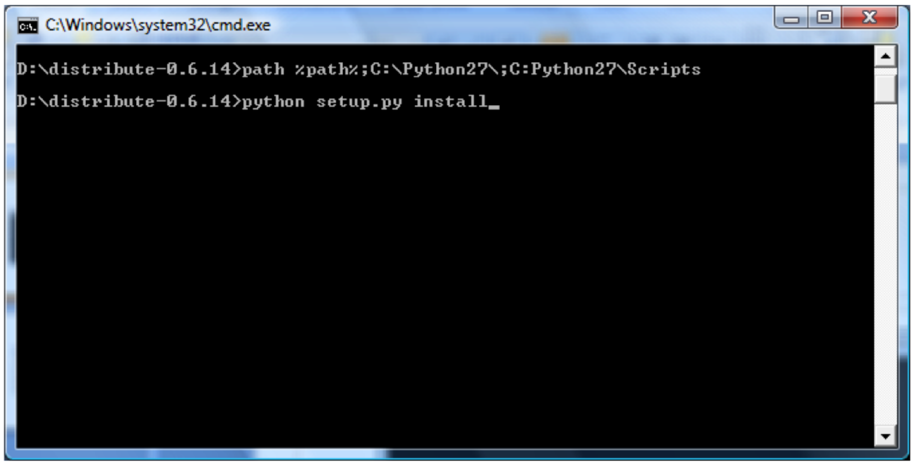
2.2 Python libraries

First, get a little program that makes our life much easier for installing additional Python packages, since it automatically handles dependencies. Go to: http://pypi.python.org/pypi/distribute#downloads and download the distribute-0.6.XX.tar.gz file.

Unpack it. If your archive utility does not support the *tar.gz* format, get 7-Zip for free. Open a command prompt *(instructions here)*. We first need to expand the system search path so that shell commands find Python (Windows only, happens automatically on MacOS/Linux). For this, either type `path %path%;C:\Python27;C:\Python27\scripts` or change these settings permanently as described in *Making the PATH settings permanent under Windows*. Note that if you do not change the permanent settings, you will need to retype the above each time you open a new command prompt window and want to use Python tools.

In the command prompt, change to the "distribute" directory, which you just unpacked. Type:

```
python setup.py install
```

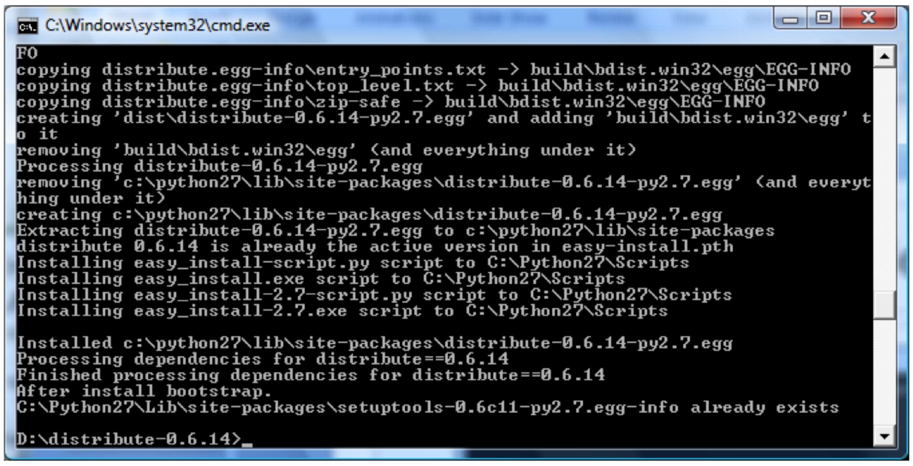


```
D:\distribute-0.6.14>path %path%;C:\Python27\;C:Python27\Scripts

D:\distribute-0.6.14>python setup.py install_
```

Note: On MacOS, you might have to prefix this line and the following `easy_install` commands by `sudo`, i.e. you need to write: `sudo python setup.py install`.

The output should look something like this:



```
FO
copying distribute.egg-info\entry_points.txt -> build\bdist.win32\egg\EGG-INFO
copying distribute.egg-info\top_level.txt -> build\bdist.win32\egg\EGG-INFO
copying distribute.egg-info\zip-safe -> build\bdist.win32\egg\EGG-INFO
creating 'dist\distribute-0.6.14-py2.7.egg' and adding 'build\bdist.win32\egg' t
o it
removing 'build\bdist.win32\egg' (and everything under it)
Processing distribute-0.6.14-py2.7.egg
removing 'c:\python27\lib\site-packages\distribute-0.6.14-py2.7.egg' (and everyt
hing under it)
creating c:\python27\lib\site-packages\distribute-0.6.14-py2.7.egg
Extracting distribute-0.6.14-py2.7.egg to c:\python27\lib\site-packages
distribute 0.6.14 is already the active version in easy-install.pth
Installing easy_install-script.py script to C:\Python27\Scripts
Installing easy_install.exe script to C:\Python27\Scripts
Installing easy_install-2.7-script.py script to C:\Python27\Scripts
Installing easy_install-2.7.exe script to C:\Python27\Scripts

Installed c:\python27\lib\site-packages\distribute-0.6.14-py2.7.egg
Processing dependencies for distribute==0.6.14
Finished processing dependencies for distribute==0.6.14
After install bootstrap.
C:\Python27\Lib\site-packages\setuptools-0.6c11-py2.7.egg-info already exists

D:\distribute-0.6.14>_
```

Now we can easily install more libraries. The only prerequisite for installing the scientific tools is `nose`, `ipython` is very helpful as well. Type:

```
easy_install nose==0.11.4
easy_install ipython
easy_install coverage
easy_install sphinx
easy_install elixir
```

Note: You can `easy_install` more packages even if you do not have administration rights by instead using:

```
easy_install --user library_name
```

2.3 Scientific Tools for Python

Now you are set to install the scientific Python tools, which you cannot `easy_install` from Python itself (for those interested – this is because these scientific tools make use of highly optimised code written in C and Fortran, as well as other libraries, which are not part of Python). So we use installers instead. The description here is geared towards Windows, for MacOS the corresponding files have a `.dmg` extension.

Start with NumPy, a package that provides matrix programming capabilities similar to those of Matlab. Go to http://sourceforge.net/projects/numpy/files/ and make sure to select the latest version for Python 2.7 (e.g. numpy-1.5.1-win32-superpack-python2.7.exe) and the correct version of the architecture (win32 if in doubt). Run the installer, accepting all default options. Somewhere along the process, you should be able to see where Python got installed – make sure that it is located in `C:\Python27` – else remember the value for your system and use it below as appropriate. In order to verify whether it installed correctly, start an interactive Python session and type the following:

```
>>> import numpy
>>> numpy.test()
```

Now download SciPy, a package with tools for scientific computations, such as optimisers, from http://sourceforge.net/projects/scipy/files/. The process should be very similar to the one before, again be sure to select the latest version and the correct one for your system. Install it, accepting all default options along the way. In order to verify whether it installed correctly, start an interactive Python session and type the following:

```
>>> import scipy
>>> scipy.test()
```

A useful extension to SciPy is scikits.statsmodels, providing some additional statistical and econometric functionality. Direct your browser to https://launchpad.net/statsmodels/+download and select the latest version (Something like scikits.statsmodels-0.3.0dev2013.zip). Unzip it, and *open a Windows command-line* (a Terminal window on the Mac) and go to the unzipped directory. Type:

```
python setup.py install
```

Things should finish with something like "Finished processing dependencies for scikits.statsmodels==0.3.0dev".

Now install matplotlib, which provides powerful plotting facilities, very similar to those of Matlab. This can be tedious for Mac users who installed a 64-bit Python (likely the default), in that case go to *Installing matplotlib (64-bit) on MacOS* (but first check whether a 64-bit dmg has been released on the download page). Else, get matplotlib from http://sourceforge.net/projects/matplotlib/files/matplotlib/. Make sure to use the correct version and accept all default options during the installation process.

3 Windows specifics

3.1 Installing SCons on Windows

Get this file (or go to http://www.scons.org/ and click on the *Windows* link in the download section on the right-hand side). Execute it (if possible with administrator rights), accepting the default options as required. **Make sure to select version 2.1 or higher!**

3.2 Opening a shell in Windows

On Windows XP, right-click on *Start*, select *run*, type *cmd*.

If you have Windows Vista or Windows 7 it seems to be even simpler (both links via google): In the Windows Explorer, right-click on the folder you want to go to (you must be in the right panel if you have the split-screen folder view) while holding down the shift key. You'll see an extra context-sensitive menu item there: Open Command Prompt here. Just click on this menu and a command window will open with the current working directory set to the folder's actual location.

If you have never used the command line before, check this page for moving between directories, working with files, etc.

3.3 Making the PATH settings permanent under Windows

(Thanks to Thomas Behles for providing the basis for this script)

You will need local administration rights for this again.

Right-click on Computer. Then go to "Properties" and select the tab "Advanced System settings". Choose "Environment Variables" and select "Path" from the list of system variables.

Choose "Edit" and **append** `;C:\Python27;C:\Python27\scripts` to the variable value – make sure the rest remains as it is.

Click on `OK` as often as needed.

Test this by opening a new command prompt in a directory that is not `C:\Python27` and type `python`. If the Python interpreter launches, it worked.

If you are using Stata, you may as well append `;C:\Path\to\Stata.exe` to the path in the same fashion now (e.g. if your Stata executable is `C:\Programme\Stata10\WSESTATA.EXE`, you should add `;C:\Programme\Stata10` to your path). Same for Matlab.

4 MacOS 10.6 specifics

4.1 Installing SCons on MacOS

Assuming you followed the steps in *Python libraries*, open a Terminal window and type:

```
sudo easy_install scons
```

If you use Stata, you should append:

```
# Stata directory
PATH="${PATH}:/Applications/Stata/StataMP.app/Contents/MacOS/"
# Finish
export PATH
```

to the .bash_profile or .profile file in your home directory. Check which one is appropriate by typing:

> ls -a ~/ | grep profile

in a Terminal session – whatever comes first of the two is used. In the path/to/Stata/ you may need to replace bits and pieces as appropriate for your installation (e.g. you might not have StataMP but StataSE).

4.2 Installing matplotlib (64-bit) on MacOS

Get the MacOS developer tools (XCode) from http://developer.apple.com/technologies/xcode.html. Click on "Mac Dev Center" on the right, and follow the subsequent steps (you will need to register first with Apple as a developer).

Check whether you have a Fortran compiler by typing `which gfortran` in a Terminal session, else get this one from http://r.research.att.com/tools/.

Download and install MacPorts.

Start a Terminal session and type:

```
sudo port install pkgconfig
sudo port install freetype
sudo port install libpng
```

Get the Matplotlib tarball from http://sourceforge.net/projects/matplotlib/files/matplotlib. Unpack the file and change to the resulting directory in a Terminal session (e.g. `cd Downloads/matplotlib-1.0`). Type:

```
python setup.py install
```

In order to verify whether it installed correctly, type `python` to get an interactive Python session and the following (the first step will take some time the first time you do this):

```
>>> from matplotlib import pyplot
>>> pyplot.plot([1,2,3], [3,8,7])
```

Done!

5 Patching SCons for use with Sphinx

Update: If you install SCons 2.1 or higher, this is not necessary anymore!!!

There is a bug in the SCons LaTeX builder which makes it not play nicely in combination with Sphinx and VariantDir. There is a patch for this, but it has not made it into the latest release yet, so you'll need to apply it yourself.

Find the file tex.py in the Tools folder of your SCons installation. On Windows, it will usually be:

```
C:\Python27\Lib\site-packages\scons-2.0.1\SCons\Tool
```

On the Mac, it is probably:

```
/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/site-packages/ [...]
scons-2.0.1-py2.7.egg/scons-2.0.1/SCons/Tool
```

Open the file `tex.py` in Eclipse (Windows) or some editor (Mac OS because you'll need administration rights to modify things you easy_installed and Eclipse can't get those. You could use, e.g., IDLE, the one supplied with Python).

Assuming you have SCons version 2.0.1 installed, line 671 of `tex.py` should read:

```
if theSearch:
```

Replace that line by:

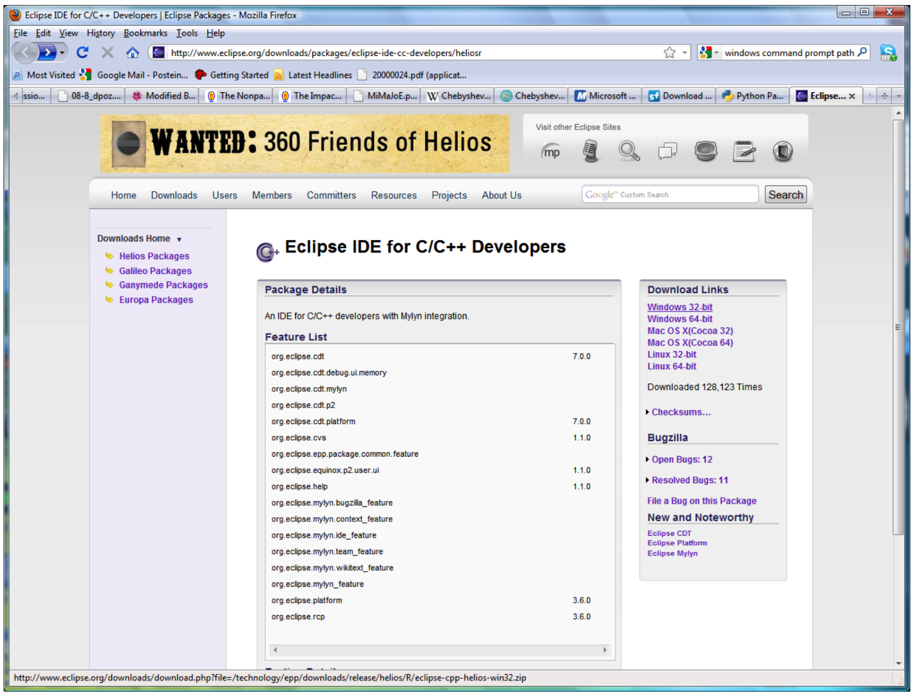
```
# Add side effects if feature is present.
# If file is to be generated, add all side effects.
if (theSearch != None) or (not source[0].exists() ):
```

6 Eclipse

6.1 Download and installation

Note: Eclipse is written in Java and requires that a Java Runtime Environment (JRE) be installed on your machine to run. On some older versions of Windows XP, there might not be an appropriate JRE installed. If you have any trouble running Eclipse, you can download one from this page. Look for "Download JRE" and follow all necessary steps.

For installing Eclipse, go to the download section of the Eclipse homepage and get the version for C/C++ developers for your system. Unpack it, move the entire folder to `C:\`, and make a shortcut to the application eclipse that you find in the folder on your desktop (or wherever you like).



Eclipse IDE for C/C++ Developers

Package Details

An IDE for C/C++ developers with Mylyn integration.

Feature List

org.eclipse.cdt	7.0.0
org.eclipse.cdt.debug.ui.memory	
org.eclipse.cdt.mylyn	
org.eclipse.cdt.p2	
org.eclipse.cdt.platform	7.0.0
org.eclipse.cvs	1.1.0
org.eclipse.epp.package.common.feature	
org.eclipse.equinox.p2.user.ui	1.1.0
org.eclipse.help	1.1.0
org.eclipse.mylyn.bugzilla_feature	
org.eclipse.mylyn.context_feature	
org.eclipse.mylyn.ide_feature	
org.eclipse.mylyn.team_feature	
org.eclipse.mylyn.wikitext_feature	
org.eclipse.mylyn_feature	
org.eclipse.platform	3.6.0
org.eclipse.rcp	3.6.0

Download Links

Windows 32-bit
Windows 64-bit
Mac OS X(Cocoa 32)
Mac OS X(Cocoa 64)
Linux 32-bit
Linux 64-bit

Downloaded 128,123 Times

Checksums...

Bugzilla

Open Bugs: 12

Resolved Bugs: 11

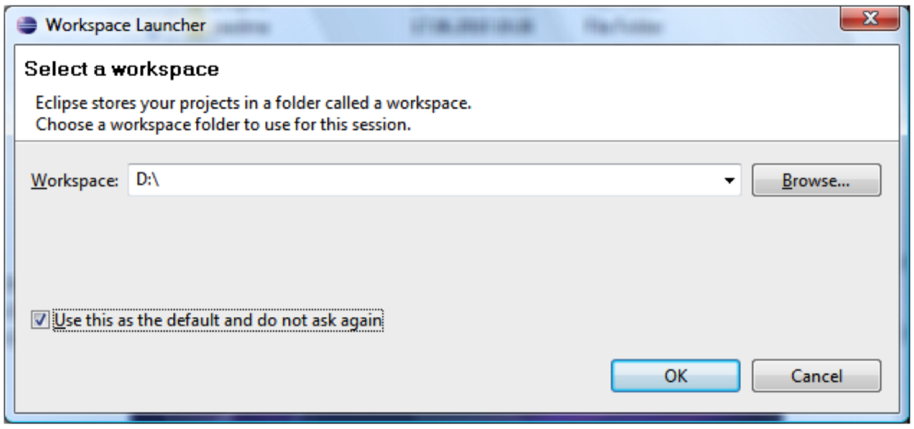
File a Bug on this Package

New and Noteworthy

Eclipse CDT
Eclipse Platform
Eclipse Mylyn

http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/helios/R/eclipse-cpp-helios-win32.zip

Start Eclipse. The first thing you need to pick is a workspace. Eclipse has a hierarchical internal structure and you will only be able to create projects in or below this directory. Try to pick or create one that does not contain any spaces – in theory it shouldn't matter, but you never know... E.g., if you have all your projects on `D:\`, just pick that:


Workspace Launcher
Select a workspace
Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.
Workspace: D:\
Browse...
☑ Use this as the default and do not ask again
OK
Cancel

Tick the box *Use this as the default ...* and click OK.

Eclipse is a modular system, and the components we need are not installed by default. We need two distinct components: The Subversion bindings for Eclipse, called *Subversive*, and the tools for using Eclipse together with Python, called *PyDev*.

6.2 Configuration of Subversive

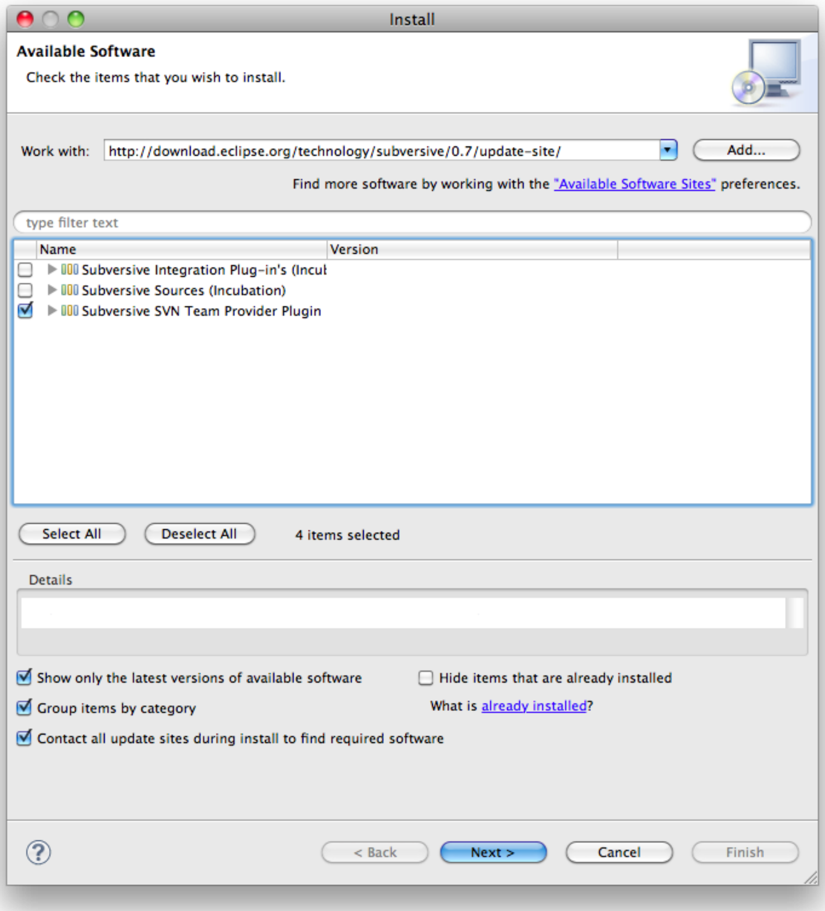
The version control system we will start to use in the next lecture is called Subversion, and it integrates directly with Eclipse.

In the menu bar, click on *Help -> Install new software*. In the address bar, type or paste the following line and press return:

```
http://download.eclipse.org/technology/subversive/0.7/update-site/
```

Select:

```
Subversive SVN Team Provider (Incubation).
```


Install

Available Software

Check the items that you wish to install.

Work with: http://download.eclipse.org/technology/subversive/0.7/update-site/ Add...

Find more software by working with the "Available Software Sites" preferences.

type filter text

	Name	Version
☐	Subversive Integration Plug-in's (Incut	
☐	Subversive Sources (Incubation)	
☑	Subversive SVN Team Provider Plugin	

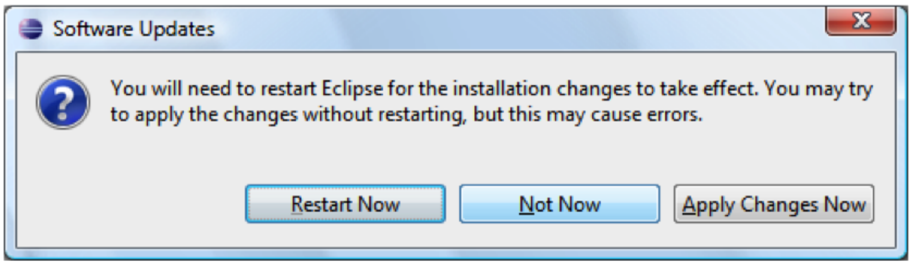
Select All Deselect All 4 items selected

Details

☑ Show only the latest versions of available software
☐ Hide items that are already installed
☑ Group items by category
What is already installed?
☑ Contact all update sites during install to find required software

< Back Next > Cancel Finish

Click on *Next* as required, accept the license terms, click on *Finish*. Do **not** restart yet, click on **not now**.


Software Updates

You will need to restart Eclipse for the installation changes to take effect. You may try to apply the changes without restarting, but this may cause errors.

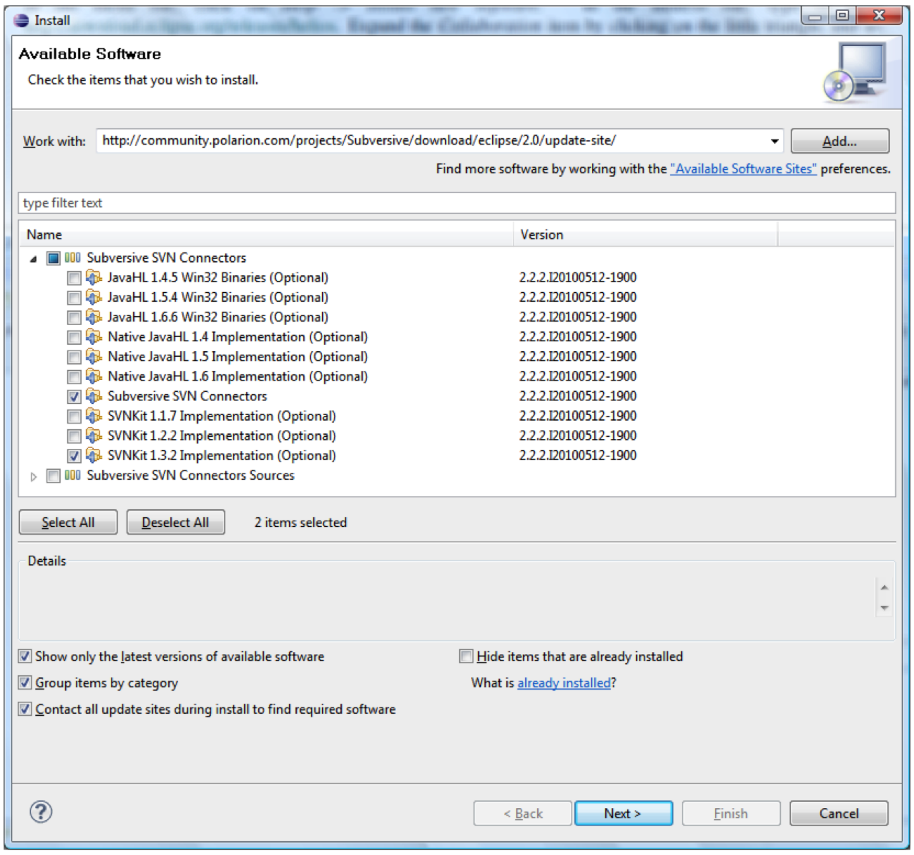
Restart Now Not Now Apply Changes Now

Once more, go to *Help* -> *Install new software* and type or paste:

```
http://community.polarion.com/projects/subversive/download/eclipse/2.0/helios-site/
```

in the address bar. Expand the *Subversive SVN Connectors* item and select both:

```
Subversive SVN Connectors
SVNKit 1.3.5 Implementation (Optional).
```


Install

Available Software

Check the items that you wish to install.

Work with: http://community.polarion.com/projects/Subversive/download/eclipse/2.0/update-site/

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
☐ Subversive SVN Connectors	
☐ JavaHL 1.4.5 Win32 Binaries (Optional)	2.2.2.I20100512-1900
☐ JavaHL 1.5.4 Win32 Binaries (Optional)	2.2.2.I20100512-1900
☐ JavaHL 1.6.6 Win32 Binaries (Optional)	2.2.2.I20100512-1900
☐ Native JavaHL 1.4 Implementation (Optional)	2.2.2.I20100512-1900
☐ Native JavaHL 1.5 Implementation (Optional)	2.2.2.I20100512-1900
☐ Native JavaHL 1.6 Implementation (Optional)	2.2.2.I20100512-1900
☑ Subversive SVN Connectors	2.2.2.I20100512-1900
☐ SVNKit 1.1.7 Implementation (Optional)	2.2.2.I20100512-1900
☐ SVNKit 1.2.2 Implementation (Optional)	2.2.2.I20100512-1900
☑ SVNKit 1.3.2 Implementation (Optional)	2.2.2.I20100512-1900
☐ Subversive SVN Connectors Sources	

2 items selected

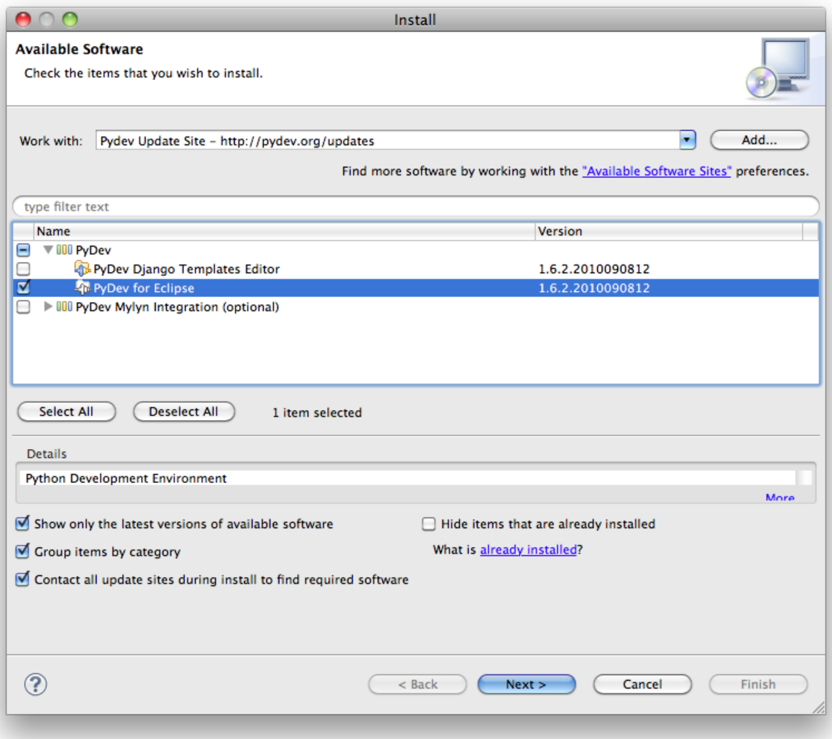
Details

☑ Show only the latest versions of available software
☐ Hide items that are already installed
☑ Group items by category
What is already installed?
☑ Contact all update sites during install to find required software

Click on *Next* as required, accept the license terms, click on *Finish*, don't mind the unsigned content, and agree to restart Eclipse.

6.3 Configuration of PyDev

Make sure you *installed Python* before you proceed. If you have installed Python, go to *Help -> Install new software* in the menu bar again. In the address bar, type http://pydev.org/updates. Expand the triangle next to *PyDev* and tick the box next to *PyDev for Eclipse*:


Install

Available Software

Check the items that you wish to install.

Work with: Pydev Update Site - http://pydev.org/updates

Find more software by working with the "Available Software Sites" preferences.

Name	Version
[–] PyDev	
[ ] PyDev Django Templates Editor	1.6.2.2010090812
[✓] PyDev for Eclipse	1.6.2.2010090812
[ ] PyDev Mylyn Integration (optional)	

1 item selected

Details

Python Development Environment

[✓] Show only the latest versions of available software

[ ] Hide items that are already installed

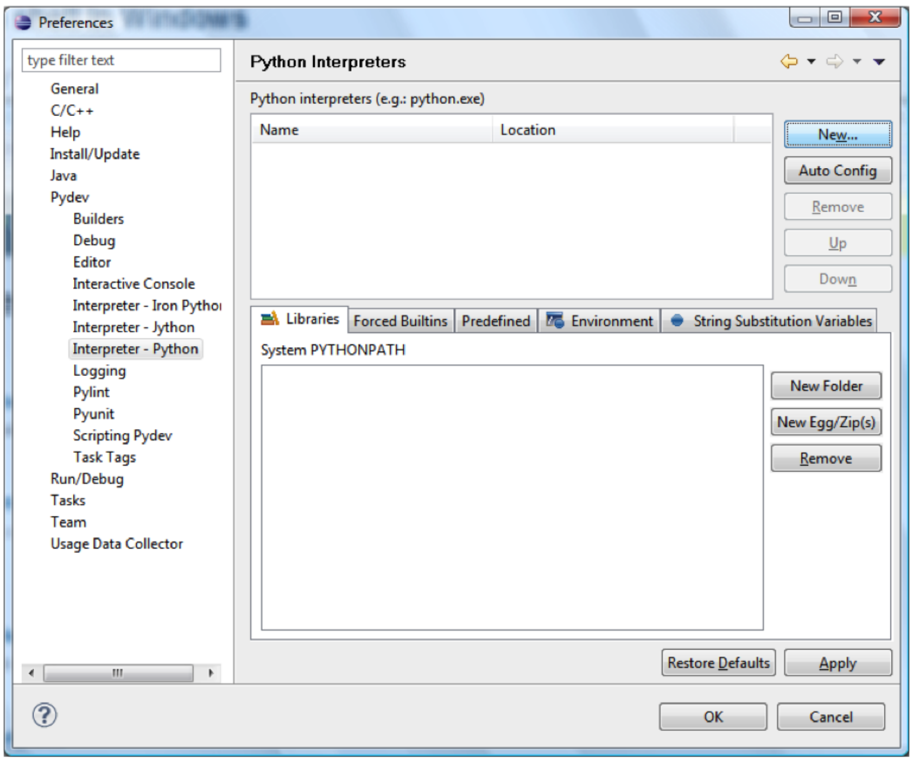
[✓] Group items by category

What is already installed?

[✓] Contact all update sites during install to find required software

Be sure **not** to select the Django Template Editor, or the install will fail. Click on *Next* as required, accept the license terms, click on *Finish*, trust all certificates, and agree to restart Eclipse.

Now we need to tell Eclipse where to find the Python interpreter. In the Eclipse menu bar, go to *Window -> Preferences* and select *PyDev -> Interpreter - Python*


Preferences

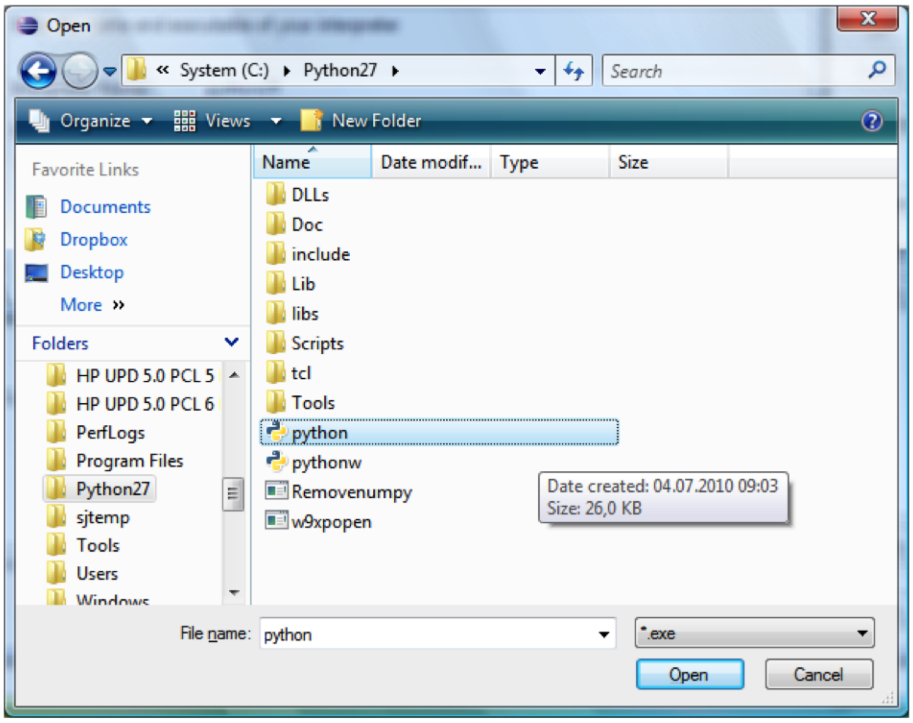
Python Interpreters

Python interpreters (e.g.: python.exe)

Name	Location

System PYTHONPATH

Click on *New...*, name the interpreter python27, and click on browse. Go to `C:\Python27\` and select python.exe:


Open

System (C:) ▸ Python27 ▸

Name | Date modif... | Type | Size

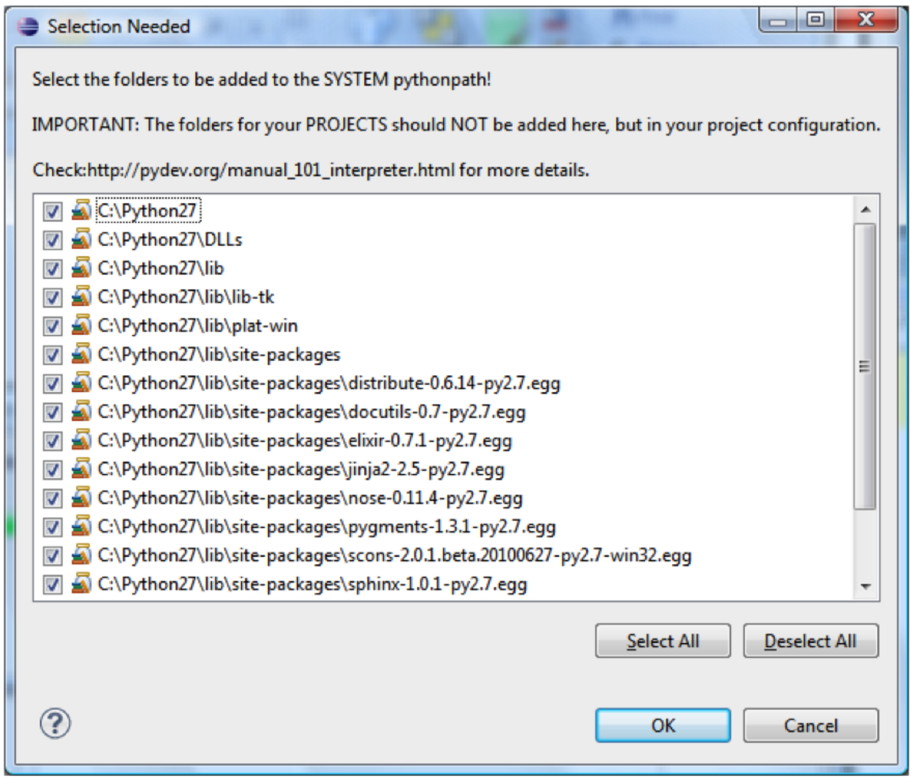
DLLs
Doc
include
Lib
libs
Scripts
tcl
Tools
python
pythonw
Removenumpy
w9xpopen

Date created: 04.07.2010 09:03
Size: 26,0 KB

File name: python

*.exe

After clicking okay, you will be asked about the SYSTEM pythonpath and hopefully you will see the libraries you installed before:


Selection Needed

Select the folders to be added to the SYSTEM pythonpath!

IMPORTANT: The folders for your PROJECTS should NOT be added here, but in your project configuration.

Check:http://pydev.org/manual_101_interpreter.html for more details.

- [x] C:\Python27
- [x] C:\Python27\DLLs
- [x] C:\Python27\lib
- [x] C:\Python27\lib\lib-tk
- [x] C:\Python27\lib\plat-win
- [x] C:\Python27\lib\site-packages
- [x] C:\Python27\lib\site-packages\distribute-0.6.14-py2.7.egg
- [x] C:\Python27\lib\site-packages\docutils-0.7-py2.7.egg
- [x] C:\Python27\lib\site-packages\elixir-0.7.1-py2.7.egg
- [x] C:\Python27\lib\site-packages\jinja2-2.5-py2.7.egg
- [x] C:\Python27\lib\site-packages\nose-0.11.4-py2.7.egg
- [x] C:\Python27\lib\site-packages\pygments-1.3.1-py2.7.egg
- [x] C:\Python27\lib\site-packages\scons-2.0.1.beta.20100627-py2.7-win32.egg
- [x] C:\Python27\lib\site-packages\sphinx-1.0.1-py2.7.egg

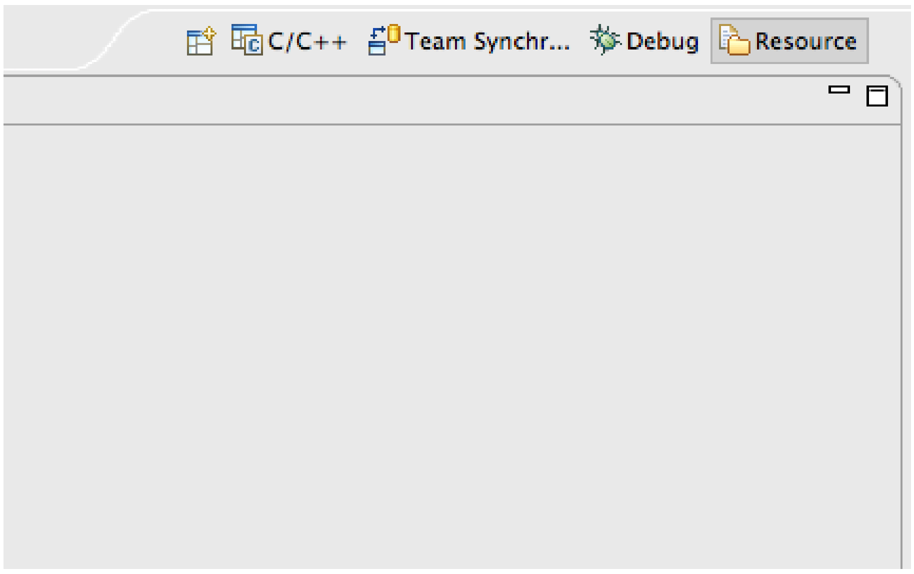
Accept the default, click okay twice. Done.

If you want to debug SCons build scripts in Eclipse, you should now follow the steps in *Making the PATH settings permanent under Windows*

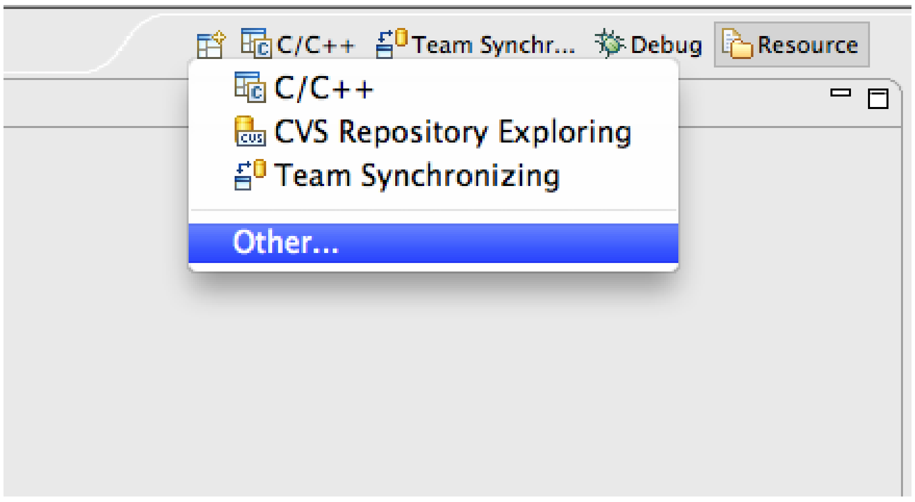
7 Troubleshooting in Eclipse

7.1 PyDev perspective not shown

Eclipse knows about various screen setups depending what you are doing at the moment. When you are writing Python code you want to see different information than when you are debugging it, etc. You can change manually between perspectives at the top right corner, and sometimes the PyDev perspective is not shown automatically, like here:



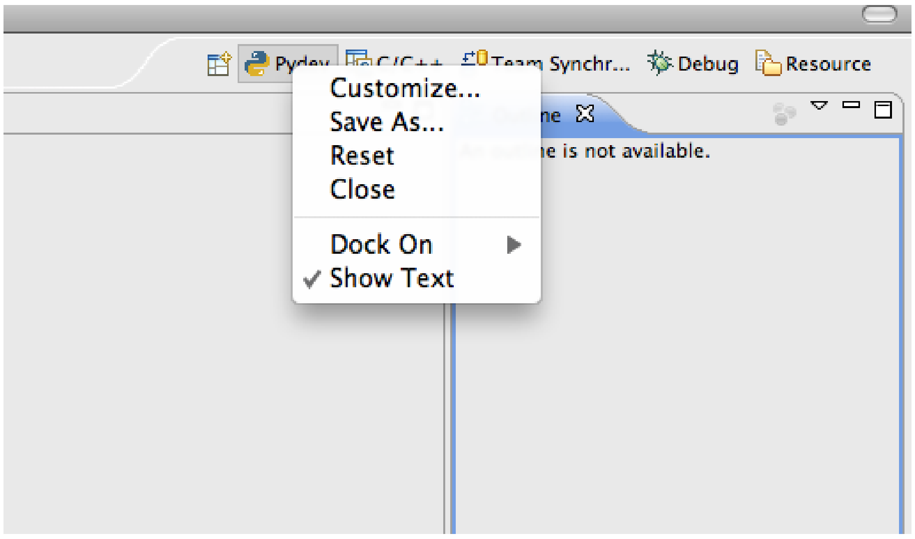
Click on the icon on the far left and select *Other...*:



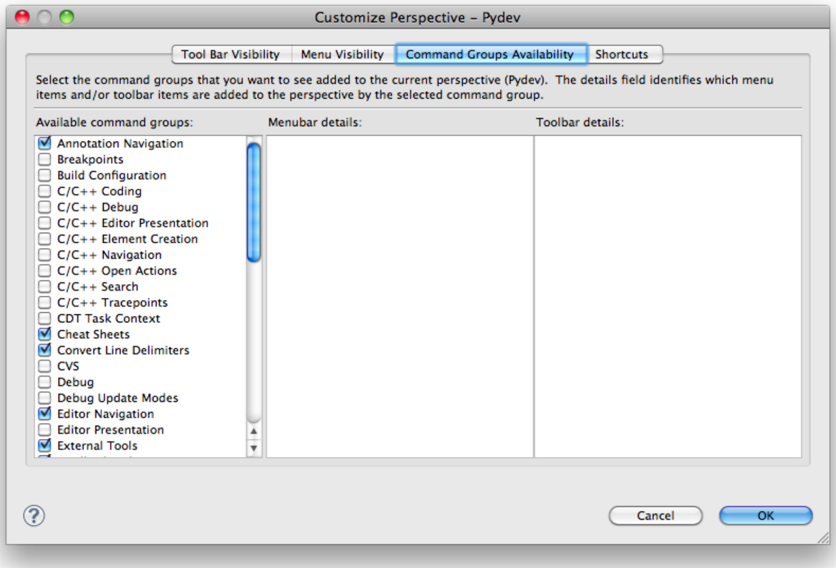
Now select PyDev and click OK.

7.2 SVN keybindings do not work

You're much faster working with keyboard shortcuts than with the mouse after you got used to it a little bit. On some installations, the keybindings to access Subversive are not associated with the PyDev perspective. To change this, right-click on the PyDev perspective at the top right corner and select *Customize...*:



Now select *Command Groups Availability* at the top of the window:



Select the command groups that you want to see added to the current perspective (Pydev). The details field identifies which menu items and/or toolbar items are added to the perspective by the selected command group.

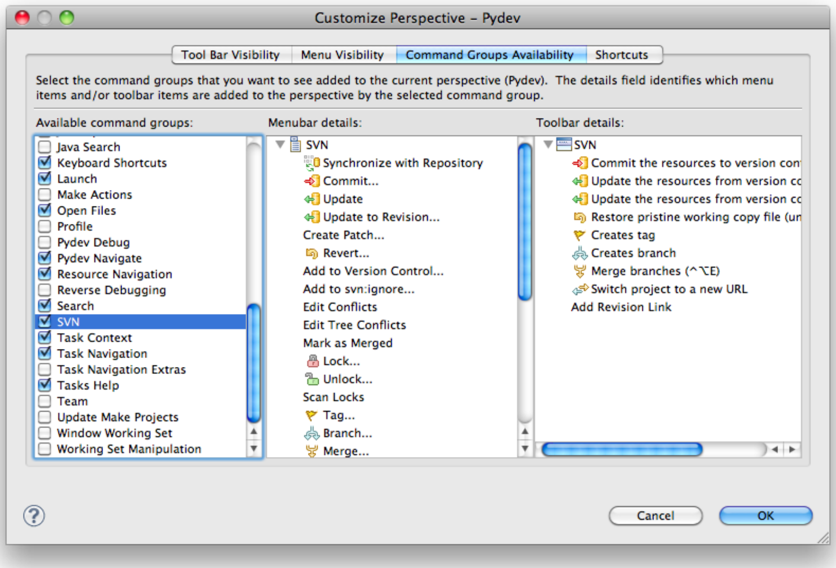
Available command groups:

- [✓] Annotation Navigation
- [ ] Breakpoints
- [ ] Build Configuration
- [ ] C/C++ Coding
- [ ] C/C++ Debug
- [ ] C/C++ Editor Presentation
- [ ] C/C++ Element Creation
- [ ] C/C++ Navigation
- [ ] C/C++ Open Actions
- [ ] C/C++ Search
- [ ] C/C++ Tracepoints
- [ ] CDT Task Context
- [✓] Cheat Sheets
- [✓] Convert Line Delimiters
- [ ] CVS
- [ ] Debug
- [ ] Debug Update Modes
- [✓] Editor Navigation
- [ ] Editor Presentation
- [✓] External Tools

Menubar details:

Toolbar details:

And check the box next to SVN in the left panel:


Customize Perspective – Pydev

Tool Bar Visibility | Menu Visibility | Command Groups Availability | Shortcuts

Select the command groups that you want to see added to the current perspective (Pydev). The details field identifies which menu items and/or toolbar items are added to the perspective by the selected command group.

Available command groups:

- [ ] Java Search
- [✓] Keyboard Shortcuts
- [✓] Launch
- [ ] Make Actions
- [✓] Open Files
- [ ] Profile
- [ ] Pydev Debug
- [✓] Pydev Navigate
- [✓] Resource Navigation
- [ ] Reverse Debugging
- [✓] Search
- [✓] SVN
- [✓] Task Context
- [✓] Task Navigation
- [ ] Task Navigation Extras
- [✓] Tasks Help
- [ ] Team
- [ ] Update Make Projects
- [ ] Window Working Set
- [ ] Working Set Manipulation

Menubar details:

```
SVN
  Synchronize with Repository
  Commit...
  Update
  Update to Revision...
  Create Patch...
  Revert...
  Add to Version Control...
  Add to svn:ignore...
  Edit Conflicts
  Edit Tree Conflicts
  Mark as Merged
  Lock...
  Unlock...
  Scan Locks
  Tag...
  Branch...
  Merge...
```

Toolbar details:

```
SVN
  Commit the resources to version con
  Update the resources from version cc
  Update the resources from version cc
  Restore pristine working copy file (ur
  Creates tag
  Creates branch
  Merge branches (^⌥E)
  Switch project to a new URL
Add Revision Link
```

Click OK. Done.

8 Setting up Eclipse to debug your SCons files

In order to set up Eclipse to work with SCons like with normal Python scripts, you need to modify a couple of settings. The reasons come from two sides: First, Eclipse is quite far detached from your system (you noticed already that the files and directories in your Eclipse Explorer are not updated automatically when you change them outside Eclipse); second, SCons is completely written in Python, but the files SCons expects do not have the `.py` extension and SCons is not started like a normal Python script.

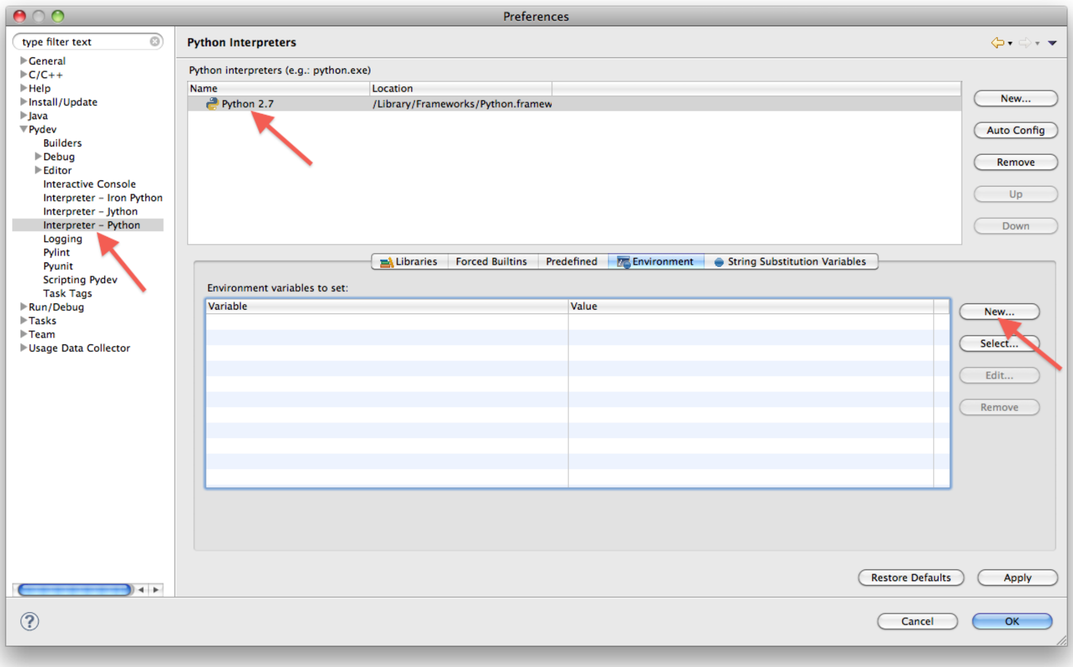
8.1 Propagating the PATH environment in Eclipse / PyDev

(This assumes you have completed *Making the PATH settings permanent under Windows* or *MacOS 10.6 specifics*, respectively)

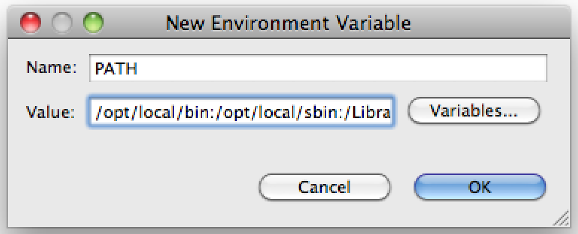
Eclipse leads a bit of a life of its own. On the one hand, this makes it a great cross-platform tool: You can use it pretty much the same on Windows, Mac, and various Linux flavours. On the other hand, it does not have access to all your system variables, in particular your system PATH. If you want to debug your SCons programs in Eclipse, you'll need to tell it about your path settings.

In the *Preferences* menu, go to:

```
PyDev -> Interpreter-Python -> Environment
```



Select *New* ..., type `PATH` in the upper field and copy your system path setting (see *Making the PATH settings permanent under Windows* under Windows, or type `echo $PATH` in a Terminal window on the Mac) to the lower field.


New Environment Variable

Name: PATH

Value: /opt/local/bin:/opt/local/sbin:/Libra

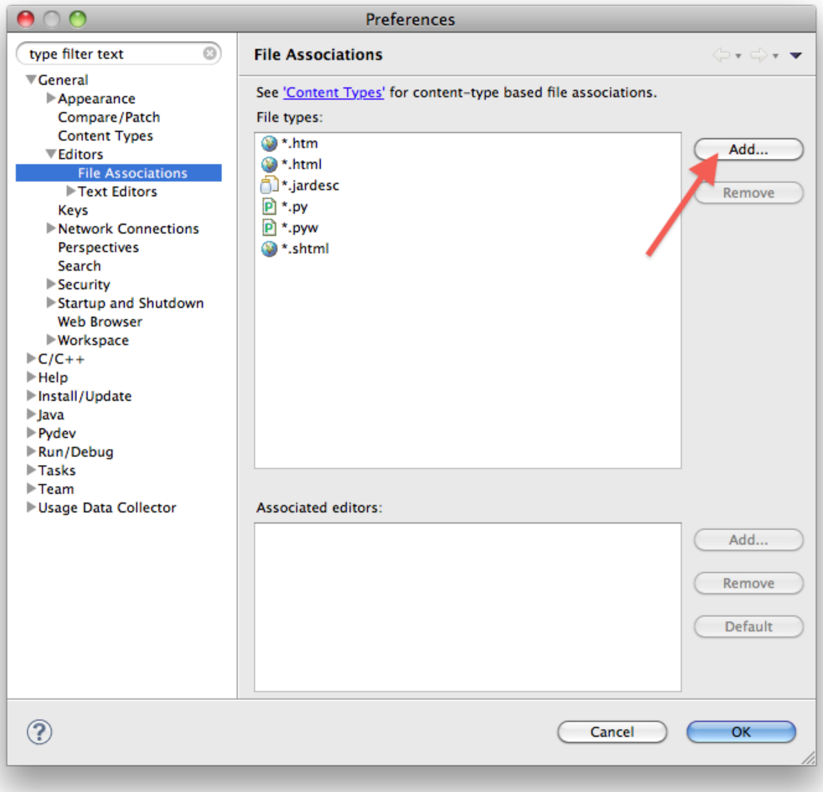
You will need to perform this step every time you update your system path, e.g. after installing a new version of Stata.

8.2 Associating SCons files with PyDev

PyDev will recognise files with the `.py` extension as Python files, but it has no way of knowing that SCons files, which are typically called SConstruct or SConscript (we will meet the latter later in the course), are also Python scripts. When you double-click on them, they will open in a standard editor window and you will not have syntax highlighting, the ability to set breakpoints, or any of the extra functionality of PyDev to work with these.

In order to change this, go to:

```
Preferences -> General -> Editors -> File Associations -> Add...
```


Preferences

File Associations

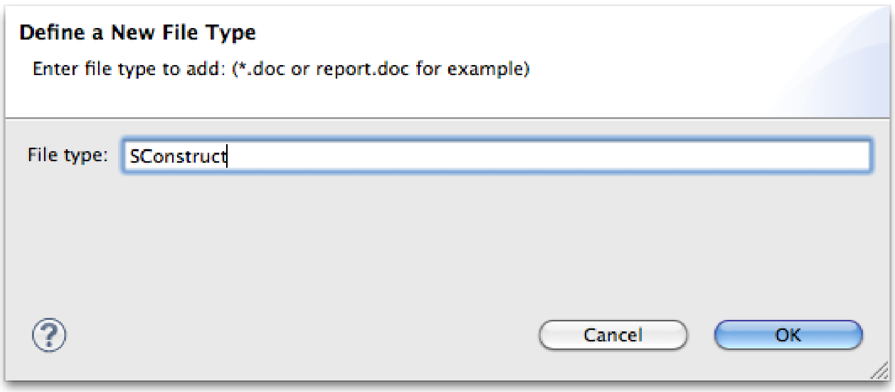
See 'Content Types' for content-type based file associations.

File types:

- *.htm
- *.html
- *.jardesc
- *.py
- *.pyw
- *.shtml

Associated editors:

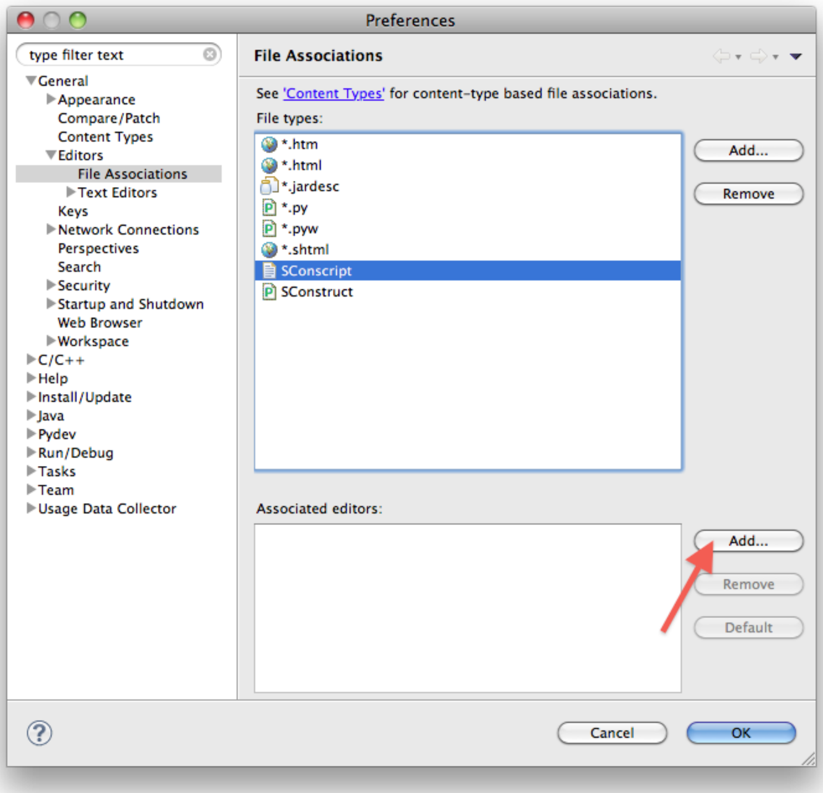
Type `SConstruct` and `OK`.


Define a New File Type

Enter file type to add: (*.doc or report.doc for example)

File type: SConstruct

Make sure that `SConstruct` is highlighted in the upper panel and select `Add ...` from the file associations panel:



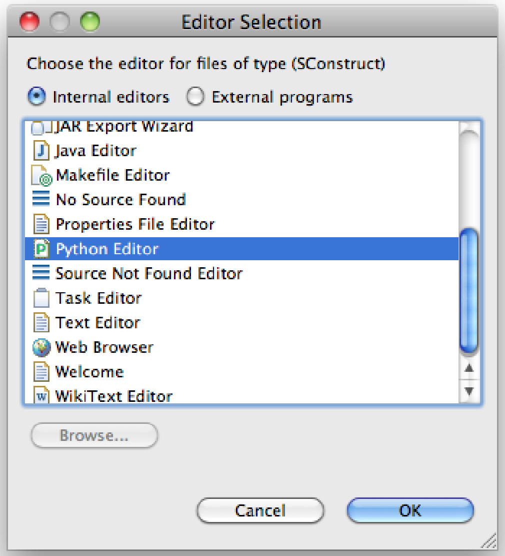
See 'Content Types' for content-type based file associations.

File types:

*.htm
*.html
*.jardesc
*.py
*.pyw
*.shtml
SConscript
SConstruct

Associated editors:

From `Internal Editors`, select `Python editor` and click `OK`.



Choose the editor for files of type (SConstruct)

[✓ Internal editors] External programs

JAR Export Wizard
Java Editor
Makefile Editor
No Source Found
Properties File Editor
Python Editor
Source Not Found Editor
Task Editor
Text Editor
Web Browser
Welcome
WikiText Editor

In the upper panel, `SConstruct` should now look the same as `*.py`. Repeat the same steps for `SConscript` and click `OK`.

8.3 Debugging SCons from Eclipse

Remember you start SCons from the command line by typing `scons` – it then looks for your SConstruct file. If the latter was a completely standard Python script, you would have needed to type `python SConstruct`. The latter is what the PyDev debugger expects, so we need to make something like this happen.

As a first step, locate the file that is executed when you type `scons` in the shell. On Windows, it will usually be `C:\Python27\scripts\scons`, on the Mac or Linux you can simply check by typing `which scons` in a Terminal window.

Now **copy** (not move) that file to the same directory where your SConstruct file lives and call it `scons.py`. You could also create a link to that file, e.g. in a Shell on MacOS or Linux, you could type:

```
ln -s /path/to/scons scons.py
```

The advantage of the latter approach is that it is automatically updated when you update SCons. In any case, make sure that the file `scons.py` is not under version control – it might be different on different machines, which would lead to trouble and also to fake changes in the repository.

In Eclipse, you can now debug your SConstruct / SConscript files by starting the debugger from the file `scons.py` as you would with any Python script.

9 Postscriptum

Last changed revision: $LastChangedRevision: 325 $

Last changed date: $LastChangedDate: 2011-02-15 09:02:52 +0100 (Tue, 15 Feb 2011) $